Estimating the Pure Characteristics Demand Model: A Computational Note

Minjae Song*

School of Economics, Georgia Institute of Technology

April, 2006

Abstract

This paper provides details of the computational algorithm of Berry and Pakes (2005) for estimating the pure characteristics demand model. The main challenge in estimating the model is to find "the mean product quality" that equates the model predicted market shares to real market shares. The algorithm combines three methods: a contraction mapping, a homotopy method with an element-by-element inversion, and Newton-Raphson method. After explaining the algorithm, I evaluate its performance using four simulated data sets with up to 20 products per market.

*minjae.song@econ.gatech.edu. Please contact the author for suggestions and comments.

I have benefited from discussions with Steve Berry, Sungkyu Lim, Aviv Nevo, and Ariel Pakes. The computer codes of the algorithm are generously provided by Steve Berry and Ariel Pakes. All errors are mine.

1 Introduction

In this paper I provide details of the computational algorithm of Berry and Pakes (2005) for estimating the pure characteristics demand model. Berry and Pakes (2005) (hereafter BP) develop a discrete choice model of differentiated product demand that does not have the idiosyncratic logit error. The model is called the pure characteristics demand model (hereafter PCM) since the consumer preference is a function of product characteristics and a price but not a function of the idiosyncratic taste. As a result, consumer heterogeneity is solely captured by random coefficients on product characteristics, and changes in consumer welfare are directly related to changes in product characteristics and a price. Song (forthcoming) uses this model to measure consumer welfare in the CPU market.

The main challenge in estimating the PCM is to find the mean product quality that equates the model predicted market shares to real market shares at given parameter values. I call this the true mean quality. Once the true mean quality is found at all possible parameter values, the subsequent estimation procedure is the same as in models with the idiosyncratic logit error. That is, the econometrician searches for parameter values that minimize the objective function of the generalized method of moments (see Berry, Levinsohn, and Pakes (1995) and Nevo (2000) for details.)

In the demand model with the idiosyncratic logit error term finding the true mean quality is conveniently done using a contraction mapping. BP shows that the market share equation in the PCM is a limiting case of the share equation in the model with the logit error term. However, the contraction mapping cannot be used for the PCM due to problems explained later.

Therefore, BP combines three methods to find the true mean quality: a contraction mapping, a homotopy method with an element-by-element inversion, and Newton-Raphson method. A basic idea is to use the contraction mapping to bring the mean quality close to the true mean quality, and then to use the other two methods repeatedly to find the true mean quality. Newton-Raphson method alone is not sufficient as the Newton step with a “bad" initial guess occasionally produces a singular or nearly singular Jacobian matrix.

The rest of the paper is organized as follows. In Section 2 I briefly explain the model and address the main challenge in the estimation procedure. In Second 3 I explain how I simulate data sets to evaluate the performance of the algorithm. In Section 4 the algorithm is explained and its performance is evaluated with the simulated data sets. I conclude with discussions in Section 5.

2 The Pure Characteristics Demand Model

The only difference between the PCM and the logit model with random coefficients (hereafter BLP) is that the latter model has the idiosyncratic logit error term in the utility function while the former does not. However, this difference makes the market share equation in the PCM very different from that in BLP. While the market share equation in BLP predicts non-zero market shares for arbitrary parameter values and any distributions of the random coefficient, the market share equation in the PCM may predict zero market shares at some parameter values.

More specifically, suppose the utility function of consumer i for product j is given as

$$u_{ij} = \overline{\delta}_j - \alpha_i p_j + \beta_{is} x_{js} \tag{1}$$

where x_{js} is a product characteristic which consumers put different values on, p_j is a price, and $\overline{\delta}_j$ is a linear function of all other product characteristics that all consumers value the same. Two random coefficients α_i and β_{is} indicate that consumer preference is heterogeneous with respect to the price and x_{js}. Depending on distributional assumptions on the random coefficients, the mean quality may be just $\overline{\delta}_j$ or include p_j and/or x_{js}. For example, suppose that both β_{is} and $\log(\alpha_i)$ are distributed normal such that $\beta_{is} \sim N(\beta_s, \sigma_s)$ and $\log(\alpha_i) \sim N(\alpha, \sigma_p)$. Then the mean quality of product j δ_j is $\overline{\delta}_j + \beta_s x_{js}$, and parameters to be estimated are $\boldsymbol{\theta} = \{\alpha, \sigma_p, \sigma_s\}$.

To derive the market share equation, products should be ordered by the price such that product 1 is the product with the lowest price and product J is the price with the highest price. Given products $j = 1, .., J$ and the outside option $j = 0$, a consumer endowed with α_i and β_{is} will buy product j if and only if

$$
\begin{aligned}
\alpha_i &< \min_{k<j} \frac{\delta_{ij} - \delta_{ik}}{p_j - p_k} \equiv \overline{\Delta}_{ij} \text{ and} \\
\alpha_i &> \max_{k>j} \frac{\delta_{ik} - \delta_{ij}}{p_k - p_j} \equiv \underline{\Delta}_{ij},
\end{aligned} \tag{2}
$$

where $\delta_{ij} = u_{ij} - \alpha_i p_j = \overline{\delta}_j + \beta_{is} x_{js}$. Given $\overline{\Delta}_{ij}$, $j = 1, ..., J-1$, $\overline{\Delta}_{i0} = \infty$, and $\overline{\Delta}_{iJ} = 0$, the market share for product j is given by

$$
s_j\left(\boldsymbol{\delta}, \mathbf{p}; \boldsymbol{\theta}, F, G\right) = \int \left(F\left(\overline{\Delta}_{ij} | \beta_{is}\right) - F\left(\underline{\Delta}_{ij} | \beta_{is}\right)\right) 1\left[\overline{\Delta}_{ij} > \underline{\Delta}_{ij}\right] dG\left(\beta_{is}\right) \tag{3}
$$

where $F\left(.|\beta_{is}\right)$ is the cdf of α_i given β_{is}, $dG\left(\beta_{is}\right)$ is the cdf of β_{is}, and $1[.]$ is the indicator function. This equation shows the model predicted market share becomes zero whenever the indicator function does not hold true for any consumers.

The main challenge in estimating the PCM is to find the true mean quality that equates real market shares to the model predicted market shares. Once the true mean quality is found, the rest of the estimation procedure is exactly the same as in BLP. For this reason I focus on the search algorithm for finding the true mean quality.

3 Data Simulation

I limit the exposition to the case where a true model is represented by the following utility function.

$$u_{ij} = \beta_0 + \beta_1 x_{1j} + \beta_{i2} x_{2j} + \xi_j - \alpha_i p_j \tag{4}$$

where $\log(\alpha_i) \sim N(0, \theta_1)$ and $\beta_{i2} \sim N(\beta_2, \theta_2)$. x_{1j} and x_{2j} are product characteristics. p_j is a product price and ξ_j is a characteristic consumers observe but the econometrician does not. There are two random coefficients: one on the price variable and the other on x_{2j}. Letting $\delta_j = \beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j} + \xi_j$,

$$u_{ij} = \delta_j + \theta_2 \upsilon_i x_{2j} - \exp(\theta_1 \omega_i) p_j \tag{5}$$

where $\upsilon_i \sim N(0,1)$ and $\omega_i \sim N(0,1)$.[1]

While in the BLP model one does not have any constraints in simulating data, it is not straightforward to generate data that make all market shares non-zero in the PCM. As an example, let's consider the one random coefficient pure characteristics model where the utility function is defined as

$$u_{ij} = \delta_j - \alpha_i p_j \tag{6}$$

where $\delta_j = X_j \beta + \xi_j$. In order to have positive market shares for all products, I should have

$$\frac{\delta_J - \delta_{J-1}}{p_J - p_{J-1}} < \ldots < \frac{\delta_3 - \delta_2}{p_3 - p_2} < \frac{\delta_2 - \delta_1}{p_2 - p_1} < \frac{\delta_1 - \delta_0}{p_1} \tag{7}$$

where $p_j < p_{j+1}$, $j = 1, \ldots, J$ (see Berry (1994).) The simulated product characteristics and price should

[1] Berry and Pakes (2005) simulate data based on the model with one product characteristic and two random coefficients, i.e.,

$$u_{ij} = \beta_0 + \beta_{1i} x_{1j} + \xi_j - \alpha_i p_j.$$

satisfy this equation to generate non-zero market shares.

For this reason BP first computes $\boldsymbol{\delta}$ based on simulated product characteristics and then makes price a exponential function of $\boldsymbol{\delta}$, i.e., $\mathbf{p} = \exp(\boldsymbol{\delta})/20$. As a result, $\frac{\delta_{j+1}-\delta_j}{p_{j+1}-p_j}$ decreases as j increases, which helps the model generate non-zero market shares. However, this method imposes a constraint that price is an increasing function of the mean product quality. This is always true in the one random coefficient model as shown above, but it is not necessarily true in the two random coefficient model.

Nevertheless, for the purpose of comparison I simulate two data sets following BP. More specifically,

$$\begin{aligned} \delta_j &= \beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j} + \xi_j, \\ p_j &= \exp(\delta_j)/20 \end{aligned} \tag{8}$$

where $x_{1j} \sim U[0,1]$ and $\xi_j = 0.005 \times N(0,1)$. In one data set ($Data1$) I draw x_2 from the uniform distribution, and in the other data set ($Data2$) I make x_2 a dummy variable and randomly assign 1 to a third of products. Note that this simulation implies that

$$\ln p_j = \beta_0' + \beta_1 x_{1j} + \beta_2 x_{2j} + \xi_j, \tag{9}$$

where $\beta_0' = \ln\left(\frac{1}{20}\right) + \beta_0$, and that the hedonic regression of the log of price on characteristics identifies coefficients in the utility function. However, this is unlikely to hold true with data collected from oligopolistic markets.[2]

I simulate two more data sets $Data3$ and $Data4$ without the constraint that price is an exponential function of the mean product quality. In $Data3$ x_2 is uniformly distributed and in $Data4$ it is a dummy variable. In particular I simulate them in the following way to avoid zero market shares.

I first generate non-zero market shares from the uniform distribution for each market. In order to

[2]For discussions on coefficients in the hedonic regression, see Pakes (2003).

have the sum of market shares less than 1, I multiply each random variable by $1.5/n_t$ where n_t is the number of products in market t. Then I generate prices by

$$p_j = a_0 + a_1 U[0,1] + a_2 x_{2j} + \xi_j \tag{10}$$

where $\xi_j = 0.005 \times N(0,1)$. With the simulated shares and prices I compute the mean quality $\widetilde{\delta_j}$ assuming that the model has only one random coefficient on the price variable. In the next step, I generate x_{1j} by

$$x_{1j} = \frac{\widetilde{\delta_j} - \beta_0 - \xi_j}{\beta_1}, \tag{11}$$

and generate the mean quality by

$$\delta_j = \beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j} + \xi_j \tag{12}$$

and market shares based on $\{\delta_j, p_j, x_{2j}, \theta_1, \theta_2\}_{j=1}^{J}$. Note that in these data sets coefficients of product characteristics in the hedonic regression are not necessarily the same as coefficients of product characteristics in the utility function.

For each data set I simulate four markets with 5, 10, 15 and 20 products respectively. Parameter values are shown in Table 1. Note that parameters a_0, a_1, and a_2 are not relevant to $Data1$ and $Data2$. Also note that a_2 is larger than β_2, meaning that prices are higher than the average willingness to pay for having (or not having) an extra unit of x_{2j}. This is necessary to keep price from becoming an increasing function of the mean product quality when x_{2j} is a dummy variable. Table 2 shows a sample of $Data4$ for a market with 15 products as an example.

4 Search Algorithm

The task of finding true $\boldsymbol{\delta}$ is a root finding problem with n_t nonlinear systems of equations. In other words, at given parameter values $\boldsymbol{\theta}$, true $\boldsymbol{\delta}$ is $\boldsymbol{\delta}$ that satisfies

$$|S_t(\boldsymbol{\delta}, \boldsymbol{\theta}, ...) - s_t| < tol, \tag{13}$$

where tol is a very small number. Newton-Raphson method is a good candidate to solve this problem. Yet, this method can fail when the Newton step produces a singular or nearly singular Jacobian matrix.[3] And it is more likely to happen as the number of products increases.

The search algorithm in BP consists of three steps. The first step is to bring the mean quality closer to the true one by using a contraction mapping. In principle, the utility function in BLP converges to the utility function in the PCM when the idiosyncratic logit error term is scaled by a factor and that factor goes to zero. The market share function in BLP, after being scaled by the factor, is

$$s_j = \frac{\exp\left[(\delta_{ij} - \alpha_i p_j)\,\mu\right]}{1 + \sum_{m=1}^{J} \exp\left[(\delta_{im} - \alpha_i p_m)\,\mu\right]} \tag{14}$$

where μ is the inverse of the scale factor. If μ can be increased to a very large number without blowing up the exponential function, the contraction mapping can be used to find true $\boldsymbol{\delta}$. However, the exponential function easily blows up before $\boldsymbol{\delta}$ approaches to true $\boldsymbol{\delta}$. With the simulated data it often happens before μ reaches 200. Nevertheless, $\boldsymbol{\delta}$ obtained by the contraction mapping with a large value of μ should be very close to true $\boldsymbol{\delta}$. Therefore, it is a good initial point for the next step of the search.

The second step is a homotopy method with an element-by-element inversion. The element-by-

[3] See Press (1992) for details.

element inversion finds a single δ_j that satisfies

$$\left|S_{jt}\left(\delta_j, \boldsymbol{\delta}_{-j}, \boldsymbol{\theta}, ...\right) - s_{jt}\right| < tol \tag{15}$$

given values of $\boldsymbol{\delta}_{-j}$. BP is not able to prove that this leads to a weak contraction with modulus strictly less than one. Instead, they combine this with a variant of the homotopy method. This method repeatedly uses the element-by-element inversion, but it uses

$$\delta_j'\left(t\right) = \left(1-t\right)\delta_{0j} + t\delta_j,\ j = 1, ..., J \tag{16}$$

as the starting point for the next iteration, where δ_{0j} is an initial guess, δ_j is a solution of the current iteration, and $0 < t \leq 1$. Note that with $t = 1$ this is a simple element-by-element inversion without the homotopy method. For $t < 1$ this is a strict contraction mapping but the fixed point is not necessarily the solution to equation (13). BP suggests that t should move very slowly to 1. For example, t may start at 0.99 and increase by 0.0025. And at a given value of t the algorithm repeats the element-by-element inversion method for 50 times before increasing t.

The third step is, if true $\boldsymbol{\delta}$ is not found yet, to use Newton-Raphson method. The algorithm moves back and forth between the second and the third step. More specifically, the algorithm uses the element-by-element inversion method repeatedly for r times at a given value of t. If equation (13) is not satisfied but all of the model predicted market shares are non-zero after r repetitions, it uses Newton-Raphson method. If equation (13) is not satisfied and some of the model predicted market shares are zero, it moves to the next value of t. Even in the Newton step, the algorithm checks if any model predicted share is zero in every iteration. If it happens, the algorithm quits the Newton loop and moves back to the second step with the next value of t. As the number of products increases, it is necessary to increase t more slowly toward 1. The whole search algorithm ends when it finds the mean quality that satisfies equation (13). When t reaches 1 without finding true $\boldsymbol{\delta}$, the search is unsuccessful.

I first fix parameter values at the true values and search for the true mean quality for each data set. There are four markets in each data set with 5, 10, 15, and 20 products respectively. Table 3 shows the mean squared error (MSE) between the mean quality found by the algorithm and the mean quality generated in the data sets, and the computational time to complete the whole search. The IBM-compatible desktop computer with 3 gigahertz Intel Pentium 4 processor and 1 gigabyte memory is used for the search. The table shows that the search algorithm finds the true mean quality accurately for all four data sets. The computational time is less than 2 minutes when x_{2j} is a continuous variable and less than 3 minutes when it is a dummy variable.

In almost all cases the homotopy method alone does not find true $\boldsymbol{\delta}$. It finds $\boldsymbol{\delta}$ that almost, but not completely, satisfies equation (13) . Then the algorithm uses it as a starting point for Newton-Raphson method. When the number of products is less than 15, the homotopy almost always makes all model predicted market shares non-zero after first 50 repetitions at an initial t value, and Newton-Raphson method finds true $\boldsymbol{\delta}$ without producing a singular Jacobian matrix. When the number of products is 15 and 20, the homotopy goes over multiple t values to make all model predicted market shares non-zeros.

Next, I run the search algorithm for 25 pairs of parameter values with $\theta_1 = \{0.6, 0.8, 1, 1.2, 1.4\}$ and $\theta_2 = \{0.6, 0.8, 1, 1.2, 1.4\}$ for $Data1$ and $Data2$ and $\theta_1 = \{0.4, 0.6, 0.8, 1, 1.2\}$ and $\theta_2 = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ for $Data3$ and $Data4$. Table 4 shows how many pairs the search algorithm finds the true mean quality for, and the computational time for each data set. The search algorithm finds the true mean quality for all parameter values in all four data sets. It takes less than an hour with the first three data sets and 1 hour and 3 minutes with $Data4$. The comparison with Table 3 suggests that the computational time when the parameter values are not true is not necessarily longer than when the parameter values are true.

5 Discussions

In this paper I explain how the algorithm in BP searches for the mean quality that equates the model predicted market shares with real market shares, and show that it works well and reasonably fast with various simulated data sets.

All three steps of the algorithm consist of essential components. In almost all cases the algorithm resorts to the Newton method to find the true mean quality, although I have a few exceptions where the first two steps of the algorithm are sufficient. However, all exceptions have only 5 products in a market. The homotopy method, the second step of the algorithm, plays an important role in making all model predicted market shares non zeros when the number of products are larger than 10.

The contraction mapping is also critical for the algorithm to find the true mean quality successfully and to search fast. For example, as an alternative starting point I use the mean quality in the vertical model, in which x_{2j} is equal to zero. When I use this starting point with $Data4$ for the same 25 pairs of the parameter values, the algorithm takes about 4 hours to complete the search, and it fails to find the true mean quality for 7 pairs.

References

[1] Berry, S. (1994), "Estimating discrete-choice models of product differentiation," *RAND Journal of Economics*, 25, 242-262.

[2] Berry, S., J. Levinsohn, and A. Pakes (1995), "Automobile Prices in Market Equilibrium," *Econometrica*, 63, 841-890.

[3] Berry, S. and A. Pakes (2005), "The Pure Characteristics Demand Model," mimeo, Harvard University.

[4] Nevo, A. (2000), "A Practitioner's Guide to Estimation of Random Coefficients Logit Models of Demand," *Journal of Economics & Management Strategy*, 9, 513-548.

[5] Pakes, A. (2003), "A Reconsideration of Hedonic Price Indices with an Application to PC's," *American Economic Review*, 93(5), 1578-1596.

[6] Press, W., S. Teukolsky, W. Vetterling, and B. Flannery (1992), *Numerical Recipes in C: the art of scientific computing*, 2nd ed., Cambridge University Press.

[7] Song, M. (forthcoming), "Measuring Consumer Welfare in the CPU Market: An application of the pure characteristics demand model", *RAND Journal of Economics*.

[Table 1] Parameter values[†]

Parameters	*Data*1 and *Data*2	*Data*3 and *Data*4
θ_1	1	0.8
θ_2	1	0.5
β_0	1	1
β_1	0.5	0.5
β_2	-0.5	-0.5
a_0	N/A	1
a_1	N/A	0.5
a_2	N/A	-0.2

[†]See the text for details on the utility functions used in simulating data.

[Table 2] Simulated data for a market with 15 products in *Data4*.[†]

	Shares	Price	Mean quality	Characteristics	
	s_j	p_j	δ_j	x_{1j}	x_{2j}
1	0.0625	1.0313	1.8099	1.5856	0
2	0.0495	1.0533	1.8424	1.6547	0
3	0.0983	1.0785	1.8753	1.7433	0
4	0.0565	1.0902	1.3884	1.7619	1
5	0.0606	1.1166	1.9146	1.8000	0
6	0.0554	1.1216	1.9191	1.8167	0
7	0.0682	1.1218	1.9193	1.8295	0
8	0.0680	1.1473	1.9362	1.8384	0
9	0.0180	1.1692	1.4497	1.8754	1
10	0.0069	1.1753	1.4531	1.8803	1
11	0.0923	1.2221	1.9770	1.9174	0
12	0.0115	1.2430	1.4870	1.9568	1
13	0.0141	1.2884	1.5040	1.9750	1
14	0.0451	1.3985	2.0451	2.0638	0
15	0.0568	1.4842	2.0706	2.1125	0

[†]In *Data4* price is not a monotonic function of the mean product quality, and one of two product characteristics is a dummy variable (see the text for details.)

[Table 3] Searching the true mean quality at the true parameter values

	$Data1^{\dagger}$	$Data2$	$Data3$	$Data4$
$MSE^{\ddagger}$	4.4646e-004	3.8791e-005	2.5903e-006	7.8567e-008
$Time^{*}$ (min:sec)	01:33	02:41	01:42	02:42

† See the text for descriptions of each data set

‡ The mean squared error between the mean quality generated in data sets and the mean quality the search algorithm finds.

* The computational time is measured with the IBM-compatible desktop computer with 3 gigahertz Intel Pentium 4 processor and 1 gigabyte memory.

[Table 4] Searching the true mean quality at 25 pairs of the parameter values

	$Data1$	$Data2$	$Data3$	$Data4$
success rate†	25/25	25/25	25/25	25/25
$Time^{*}$ (hour:min:sec)	00:44:05	00:59:01	00:48:18	01:03:20

†The number of pairs of parameter values that the search algorithm finds the true mean quality for divided by 25.

*The computational time is measured with the IBM-compatible desktop computer with 3 gigahertz Intel Pentium 4 processor and 1 gigabyte memory.